

METHOD AND APPARATUS FOR REDUCING LEAKAGE POWER IN A CACHE MEMORY USING ADAPTIVE TIME-BASED DECAY

Field of the Invention

The present invention relates generally to cache memory devices, and more particularly, to adaptive techniques for reducing the leakage power in such cache memories.

Background of the Invention

Cache memories reduce memory access times of large external memories. FIG. 1 illustrates a conventional cache architecture where a cache memory 120 is inserted between one or more processors 110 and a main memory 130. Generally, the main memory 130 is relatively large and slow compared to the cache memory 120. The cache memory 120 contains a copy of portions of the main memory 130. When the processor 110 attempts to read an area of memory, a check is performed to determine if the memory contents are already in the cache memory 120. If the memory contents are in the cache memory 120 (a cache “hit”), the contents are delivered directly to the processor 110. If, however, the memory contents are not in the cache memory 120 (a cache “miss”), a block of main memory 130, consisting of some fixed number of words, is typically read into the cache memory 120 and thereafter delivered to the processor 110.

Cache memories 120 are often implemented using CMOS technology. To achieve lower power and higher performance in CMOS devices, however, there is an increasing trend to reduce the drive supply voltage (V_{dd}) of the CMOS devices. To maintain performance, a reduction in the drive supply voltage necessitates a reduction in the threshold voltage (V_{th}), which in turn increases leakage power dissipation exponentially. Since chip transistor counts continue to increase, and every transistor that has power applied will leak irrespective of its switching activity, leakage power is expected to become a significant factor in the total power dissipation of a chip. It has been estimated that the leakage power dissipated by a chip could equal the dynamic power of the chip within three processor generations.

One solution for reducing leakage power is to power down unused devices. M.D. Powell et al., “Gated- V_{dd} : A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories,” ACM/IEEE International Symposium on Low Power Electronics and Design

(ISLPED) (2000) and Se-Hyun Yang et al., "An Integrated Circuit/Architecture Approach to Reducing Leakage in Deep-Submicron High-Performance I-Caches," ACM/IEEE International Symposium on High-Performance Computer Architecture (HPCA) (Jan. 2001), propose a micro-architectural technique referred to as a dynamically resizable instruction (DRI) cache and a gated-
 5 V_{dd} circuit-level technique, respectively, to reduce power leakage in static random access memory (SRAM) cells by turning off power to large blocks of the instruction cache.

United States Patent Application Serial Number 09/865,847, filed May 25, 2001, entitled, "Method and Apparatus for Reducing Leakage Power in a Cache Memory," incorporated by reference herein, discloses a method and apparatus for reducing leakage power in
 10 cache memories by removing the power of individual cache lines that have been inactive for some period of time assuming that these cache lines are unlikely to be accessed in the future. While the disclosed cache decay techniques reduce leakage power dissipation by turning off power to the cache lines that have not been accessed within a specified decay interval, such cache decay techniques will increase the miss rate of the cache (i.e., when a cache line is accessed that
 15 has been decayed prematurely). A need therefore exists for an adaptive method and apparatus for reducing leakage power in cache memories that adjusts the decay interval based on the performance of the cache following a cache decay.

Summary of the Invention

20 Generally, an adaptive cache decay technique is disclosed that removes power from cache lines that have not been accessed for a variable time interval, referred to as the cache line decay interval, assuming that these cache lines are unlikely to be accessed in the future. A variable cache line decay interval is established for each application or for each individual cache line. The decay interval may be increased or decreased for individual cache lines to increase
 25 cache performance or save power, respectively. In an exemplary embodiment, a default decay interval is initially established for the cache and the default decay interval may then be adjusted for a given cache line based on the performance of the cache line following a cache decay.

The cache decay performance is evaluated by determining if a cache line was decayed too quickly. For example, if a cache line is decayed and the same cache contents are
 30 again required, then the cache line was decayed too quickly and the cache line decay interval is

increased. On the other hand, if a cache line is decayed and the cache line is then accessed to obtain a different cache content, the cache line decay interval can be decreased. Thus, to evaluate the cache decay performance, a mechanism is required to determine if the same cache contents are again accessed.

5 The decay interval is maintained using a timer that is reset each time the corresponding cache line is accessed. If the interval timer exceeds the current decay interval for a given cache line, power to the cache line is removed. Once power to the cache line is removed, the contents of the data field, and (optionally) the tag field are allowed to degrade while the valid bit associated with the cache line is reset. When a cache line is later accessed after being
10 powered down by the present invention, a cache miss is incurred (because the valid bit has been reset) while the cache line is again powered up and the data is obtained from the next level of the memory hierarchy. In addition, a test is performed to evaluate the cache decay performance by determining if the same cache contents are again accessed (e.g., whether the address associated with a subsequent access is the same address of the previously stored contents). The cache decay interval is then adjusted accordingly.

The cache decay techniques of the present invention can be successfully applied to both data and instruction caches, to set-associative caches and to multilevel cache hierarchies.

A more complete understanding of the present invention, as well as further features and advantages of the present invention, will be obtained by reference to the following
20 detailed description and drawings.

Brief Description of the Drawings

FIG. 1 illustrates a conventional cache architecture;

FIG. 2 illustrates the structure of the conventional cache memory of FIG. 1 in
25 further detail;

FIG. 3 illustrates a cache memory in accordance with the cache decay techniques of United States Patent Application Serial Number 09/865,847, filed May 25, 2001, entitled, "Method and Apparatus for Reducing Leakage Power in a Cache Memory;"

FIG. 4 illustrates the structure of a cache memory in accordance with the present
30 invention;

FIGS. 5 through 7 illustrate various digital implementations of cache memories in accordance with the present invention;

FIG. 8 provides a state diagram for the exemplary two-bit counter of FIG. 5; and

FIG. 9 illustrates an analog implementation of a decay counter for a cache memory in accordance with the present invention.

Detailed Description

FIG. 2 illustrates the structure of the conventional cache memory 120 of FIG. 1 in further detail. As shown in FIG. 2, the cache memory 120 consists of C cache lines of K words each. The number of lines in the cache memory 120 is generally considerably less than the number of blocks in main memory 130. At any time, a portion of the blocks of main memory 130 resides in lines of the cache memory 120. An individual line in the cache memory 120 cannot be uniquely dedicated to a particular block of the main memory 130. Thus, as shown in FIG. 2, each cache line includes a tag indicating which particular block of main memory 130 is currently stored in the cache 120. In addition, each cache line includes a valid bit indicating whether the stored data is valid.

The present invention provides adaptive cache decay technique that remove power from cache lines that have not been accessed for a variable time interval, referred to as the cache line decay interval. Thus, the present invention allows a variable cache line decay interval to be uniquely established for each application, or even for each individual cache line associated with an application. The decay interval may adjusted for each cache line to increase performance or save power, as desired. In one exemplary embodiment, a default decay interval is initially established for the cache and the default decay interval may then be adjusted for a given cache line based on the performance of the cache line following a cache decay.

Generally, after a cache line is decayed in accordance with the present invention, the cache line performance is evaluated by determining if the cache line was decayed too quickly. For example, if a cache line is decayed and the same cache contents are required (i.e., the contents of the same block of main memory 130), the cache line was decayed too quickly and the cache line decay interval is increased. Similarly, if a cache line is decayed and the cache line is then accessed for different cache contents, the cache line decay interval is decreased. Thus, to

evaluate the cache decay performance, a mechanism is required to determine if the same cache contents are again accessed. In one embodiment, the power is maintained on the tag portion of a cache line following a decay, so that the address associated with a subsequent access can be compared to the address of the previously stored contents. When power is always maintained on the tags, there is a significant amount of power is consumed (approximately 10% of the cache leakage for a 32 KB cache). Thus, in a further variation, the present invention assumes that the same cache contents are required if a subsequent access occurs within a specified time interval. In other words, the present invention infers possible mistakes according to how fast a cache miss occurs after a cache decay.

Time-Based Cache Decay

The cache decay techniques described in United States Patent Application Serial Number 09/865,847, filed May 25, 2001, entitled, "Method and Apparatus for Reducing Leakage Power in a Cache Memory," reduce leakage power dissipation in caches. The power to a cache line that has not been accessed within a decay interval is turned off. When a cache line is thereafter accessed that has been powered down, a cache miss is incurred while the line is powered up and the data is fetched from the next level of the memory hierarchy. The recency of a cache line access is represented via a digital counter that is cleared on each access to the cache line and incremented periodically at fixed time intervals. Once the counter reaches a specified count, the counter saturates and removes the power (or ground) to the corresponding cache line.

It has been observed that decay intervals tend to be on the order of tens or hundreds of thousands of cycles. The number of cycles needed for a reasonable decay interval thus makes it impractical for the counters to count cycles (too many counter bits would be required). Thus, the number of required bits can be reduced by "ticking" the counters at a much coarser level, for example, every few thousand cycles. A global cycle counter can be utilized to provide the ticks for smaller cache-line counters. Simulations have shown that a two-bit counter for a given cache line provides sufficient resolution with four quantized counter levels. For example, if a cache line should be powered down 10,000 clock cycles following the most recent access, each of the four quantized counter levels corresponds to 2,500 cycles.

FIG. 3 illustrates a digital implementation of a cache memory 300 in accordance with United States Patent Application Serial Number 09/865,847. As shown in FIG. 3, the cache

memory 300 includes a two-bit saturating counter 320-n (hereinafter, collectively referred to as counters 320) associated with each cache line, and an N-bit global counter 310. In addition, each cache line includes a tag indicating which particular block of main memory 130 is currently stored in the cache line and a valid bit indicating whether the stored data is valid. To save power, the global counter 310 can be implemented, e.g., as a binary ripple counter. An additional latch (not shown) holds a maximum count value that is compared to the global counter 310. When the global counter 310 reaches the maximum value, the global counter 310 is reset and a one-clock-cycle T signal is generated on a global time signal distribution line 330. The maximum count latch (not shown) is non-switching and does not contribute to dynamic power dissipation. In general and on average using small cache line counters, very few bits are expected to switch per cycle.

To minimize state transitions in the counters 320 and thus minimize dynamic power consumption, the exemplary digital implementation of the present invention uses Gray coding so that only one bit changes state at any time. Furthermore, to simplify the counters 320 and minimize the transistor count, the counters 320 are implemented asynchronously. In a further variation, the counters 310, 320 can be implemented as shift registers.

For a more detailed discussion of the implementation details of the cache memory 300, see United States Patent Application Serial Number 09/865,847, filed May 25, 2001, entitled, "Method and Apparatus for Reducing Leakage Power in a Cache Memory," incorporated by reference herein.

Adaptive Time-Based Cache Decay

FIG. 4 illustrates the structure of a cache memory 400 in accordance with the present invention. As shown in FIG. 4, the cache memory 400 consists of C cache lines of K words each. Each cache line includes a tag identifying the particular block of main memory 130 that is currently stored in the cache 400 and a valid bit indicating whether the stored data is valid. In addition, in accordance with the present invention, each cache line has an associated field that records the current decay interval for the cache line. As previously indicated, the decay interval can be varied by the present invention based on cache performance following a cache decay. There are three exemplary methods discussed herein to vary the decay interval on a cache-line by cache-line basis.

In the first method, shown in FIG. 5, the current decay interval field 420 in the cache memory 400 controls the size of a local counter 520 associated with an individual cache line 550. In this case, a small decay interval only utilizes few of the bits of the local counter 520 to count the passage of time. A large decay interval utilizes more bits of the local cache line counter 520. In this embodiment, there is only one global counter 510 providing the timing signal, T. This first method requires local counters 520 of variable size and of a small number of bits that can be controlled by the decay interval field 420. The decay interval for a given cache line 550 is the result of multiplying the fixed global time period, T, by the maximum count of the local counter 520 which is set independently for a given cache line 550.

In addition, each cache line 550 includes the data, a tag indicating which particular block of main memory 130 is currently stored in the cache line 550, a valid bit (V) indicating whether the stored data is valid and a dirty bit (D) indicating whether the value stored in the cache line 550 needs to be written back to the appropriate location of main memory 130, as identified by the tag. The dirty bit is set by the processor each time the cache is updated with a new value without updating the corresponding location(s) of main memory.

To save power, the global counter 510 can be implemented, for example, as a binary ripple counter. An additional latch (not shown) holds a maximum count value that is compared to the global counter 510. When the global counter 510 reaches the maximum value, the global counter 510 is reset and a one-clock-cycle T signal is generated on a global time signal distribution line, T. The maximum count latch (not shown) is non-switching and does not contribute to dynamic power dissipation. Generally, very few bits are expected to switch per cycle, on average, using small cache line counters. The cache memory 500 shown in FIG. 5 can be part of a digital signal processor (DSP), microcontroller, microprocessor, application specific integrated circuit (ASIC) or another integrated circuit.

The second method, shown in FIG. 6, is similar to the method discussed above in conjunction with FIG. 5, but instead of using a variable sized local counter 520, a fixed sized local counter 620 of sufficient number of bits (possibly more than two) is used. A comparator 630 is then used to implement the variable maximum value that the counter is allowed to reach. The comparator 630 is set by the decay interval field 420 to a predetermined value. The local counter 620 is allowed to count up to this predetermined value. When the local counter 620

reaches the value set in the comparator it is considered the end of the count as in the previous cases.

In the third method, shown in FIG. 7, a number of different global counters N_0 through N_n (hereinafter, collectively referred to as global counters N), representing different decay intervals, are provided. The decay interval field 420 for a given cache line 550 generates a signal that is applied to a global counter selector 715 to thereby select the global counter N_i from which it will receive the timing signal to feed to the local (fixed-sized) cache-line counter 720. In this case, the decay interval 420 for a given cache line 550 is the result of multiplying the selected global time period N_i by the maximum count of the local counter 720 which is fixed for all cache lines. A small decay interval field selects the timing signal of a small global counter and a large decay interval field selects the timing signal of a large global counter. The magnitudes of the global counters are determined independently (either statically or dynamically at run-time) to suit the application or the operational environment of the computer system (low power or high performance).

In an implementation where a possible mistake is inferred based on how fast a cache miss occurs after a cache decay, the local counter 520, 620, 720 of a cache line 550 is reset upon decay and then reused to gauge dead time (i.e., the amount of time until a subsequent access). If dead time turns out to be short (e.g., the local counter did not advance a single step), then a mistake is inferred, causing a decay-miss. However, if the local counter reaches its maximum value while still in the dead period, then a successful decay is inferred. A mistake-miss with the counter at minimum value (00 or 11 in a two bit counter implementation), causes the decay interval to be adjusted upwards. A successful decay with counter at maximum value (10) causes the decay interval to be adjusted downwards. Misses with the counter at intermediate values (01 or 11) do not affect the decay interval. This implementation can be extended to the variable size counters 620, 630 mentioned above in conjunction with FIG. 6. In a variable sized counter, only events in the first value and the last value can affect the decay interval whereas events in the intermediate values have no effect.

Under different assumptions about power consumption or to improve performance or power, the range of values where the decay interval field can be increased or decreased can be modified. Three ranges of values of the effective local counter (if it is of variable size or variable

maximum count) are defined, namely, (i) the range of values where the decay interval increases, (ii) the range of values where the decay interval remains unaffected and (iii) the range of values where the decay interval is decreased. These ranges are selected to suit the computing environment and can be changed dynamically depending on the requirements of the computing system (performance or power conservation).

FIG. 8 provides a state diagram 800 for exemplary two-bit (S0, S1), saturating, Gray-code counters 520 with two inputs (WRD and decay interval (DI)). Generally, each cache line contains circuitry to implement the state machine depicted in FIG. 8. T is the global time signal generated by the (synchronous) global counter 510 to indicate the passage of time. DI is the current decay interval setting for the cache line. The second state machine input is the cache line access signal, WRD, which is decoded from the address and is the same signal used to select a particular row within the cache memory 500 (e.g., the WORD-LINE signal). As shown in FIG. 8, state transitions occur asynchronously on changes of the two input signals, DI and WRD. Since DI and WRD are well-behaved signals, there are no meta-stability problems. The only output is the cache-line switch state, PowerOFF (POOFF). The cache line is reset and returns to state 00 each time the cache line is accessed.

When power to a cache line is turned off (state 10), the cache decay should disconnect the data and (optionally) corresponding tag fields associated with the cache line from the power supply. Removing power from a cache line has important implications for the rest of the cache circuitry. In particular, the first access to a powered-off cache line should:

1. result in a cache miss (since data and tag might be corrupted without power);
2. reset the corresponding counter 520-i and restore power to the cache line (i.e., restart the decay mechanism); and
3. be delayed for a period of time until the cache-line circuits stabilize after power is restored (the inherent access time to main memory should be a sufficient delay in many situations).

To satisfy these requirements, the present invention employs the Valid bit of the cache line as part of the decay mechanism, as discussed above in conjunction with FIGS. 5-7, shown in FIG. 7. The cache-line power control in accordance with the present invention ensures that the valid bit is always powered (as is the counter). Second, a reset capability is

provided to the valid bit so it can be reset to 0 (invalid) by the decay mechanism. The PowerOFF signal clears the valid bit. Thus, the first access to a powered-off cache line always results in a miss regardless of the contents of the tag. Since satisfying this miss from the lower memory hierarchy is the only way to restore the valid bit, a newly powered cache line will have enough
 5 time to stabilize. In addition, no other access (to this cache line) can read the possibly corrupted data in the interim.

The recency of a cache line access can alternatively be implemented using an event, such as the charging or discharging of a capacitor 910, as shown in FIG. 9. Thus, each time a cache line is accessed, the capacitor is grounded. In the common case of a frequently
 10 accessed cache-line, the capacitor will be discharged. Over time, the capacitor is charged through a resistor 920 connected to V_{dd} . The bias of a voltage comparator 930 is adjusted in accordance with the present invention using the decay interval (DI). Once the charge reaches a value corresponding to the decay interval, the voltage comparator 930 detects the charge, asserts the PowerOFF signal and disconnects the power supply from the corresponding cache line.

15 It is to be understood that the embodiments and variations shown and described herein are merely illustrative of the principles of this invention and that various modifications may be implemented by those skilled in the art without departing from the scope and spirit of the invention.